

REMARKS/ARGUMENTS

Regarding Drawings

The drawings stand objected to because of margin informalities as noted in PTOL Form 948. Submitted herewith as separate papers are formal drawings in which all informalities have been corrected.

Regarding Amendments

Claims 1-5, 7-10, 12-32, 34-37, 39-59, 61-64 and 66-81 are now pending. No claims stand allowed.

Claims 6, 11, 33, 38, 60, and 65 have been cancelled, without prejudice or disclaimer.

Claims 1-5, 8-10, 13-15, 28-32, 35-37, 40-42, 55-59, 62-64, and 67-69 have been amended to further particularly point out and distinctly claim subject matter regarded as the invention, of which 1-5, 28-32, and 55-59 are independent claims. The text of claims 7, 12, 16-27, 34, 39, 43-54, 61, 66, and 70-81 is unchanged, but their meaning is changed because they depend from amended claims.

As all the independent claims have been amended to include limitations not shown or suggested by the cited art, the Examiner's objections should now be rendered moot.

The 35 U.S.C. § 102 Rejection

Claims 1-5, 7-10, 12-32, 34-37, 39-59, 61-64, and 66-81 stand rejected under 35 U.S.C. § 102 (e) as being allegedly anticipated by Wilkinson et al.^{1 2} This rejection is respectfully traversed.

MPEP § 2131 provides:

A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described in a single prior art reference.³ The identical invention must be shown in as complete detail as contained in the claim.⁴ The elements must be arranged as required by the claim.⁵

Each and every element as set forth in the present claims are not found in Wilkinson et al. Furthermore, the various combinations of elements proposed by the Office Action are never arranged by Wilkinson et al. in the same manner as proposed by the Office Action or as required by the present claims.

In the Office Action dated July 2, 2003, the Examiner stated:

¹ USP 6,308,317.

² Office Action dated July 7, 2003, ¶ 2.

³ *Verdegaal Bros. v. Union Oil Co. Of California*, 814 F.2d 628,631, 2 USPQ2d 1051,1053 (Fed. Cir. 1987).

⁴ *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913,1920 (Fed. Cir. 1989).

⁵ MPEP § 2131.

The applicant added a new feature to each of the independent claims, which indicates that there are different types (i.e. more than one) of referenceable tokens in a single package, as indicated below.⁶

The Applicants respectfully submit the Examiner has mischaracterized the amendments made to the independent claims. The independent claims were amended to indicate tokens have a type, that tokens belonging to the same type represent the same kind of referenceable item, and further that each kind of referenceable item in a package has its own independent scope for tokens of the corresponding type. Furthermore, the scope of the tokens is of limited capacity thus providing a more efficient use of a namespace, regardless of whether embodiments of the present invention are applied to a resource-constrained device or a resource-rich devices such as desktop computer. With this Amendment, the independent claims (claims 1-5, 28-32, and 55-59) have been modified to make this distinction more clear.

Claim 5

Claim 5 as amended recites:

A method for constructing an image of a first package of code comprising at least one internally referenceable item, the method comprising:
forming a mapping of said at least one internally referenceable item to an optimized numeric value having a type, optimized numeric values belonging to the same type representing the same kind of referenceable item, each kind of referenceable item in said package having its own independent scope of limited capacity for optimized numeric values of the corresponding type;
replacing references to said at least one internally referenceable item with the corresponding numeric value; and
forming the package.

⁶ Office Action ¶ 2.

The Examiner states:

Wilkinson teaches a mapping of a referenceable item to a corresponding token via his strings (referenceable items, i.e. more than one), which are mapped into integers (tokens having a numeric value), col. 9 lines 27-37. The class file (package) comprises integers (ID's) that are optimized for the specific architecture it is utilized on. The ID's (tokens having a numeric value), contrary to the applicant's assertions, are not merely single entities for the class file (package) only. The ID's, "Each ID uniquely identifies a particular object, class, field or method in the application", have their own independent scope. The applicant argues that the previous sentence indicates that a single token is used; while, the alternative form, "or", indicates multiple IDs (not a single ID. Therefore, each of the various (multiple) referenceable items in said package have their own independent scope for tokens of the corresponding token type...⁷

The Applicants respectfully disagree. Contrary to the Examiner's statement, Wilkinson et al. does not disclose forming a mapping of said at least one internally referenceable item to an optimized numeric value having a type, optimized numeric values belonging to the same type representing the same kind of referenceable item, each kind of referenceable item in said package having its own independent scope of limited capacity for optimized numeric values of the corresponding type.

Wilkinson et al. recites:

To avoid dynamic linking in the card, all the information that is distributed across several Java class files 24a, 24b, and 24c that form the application 24, are coalesced into one card class file 27 by the process shown in the flowchart in FIG. 5 ... The constant pool 42 is compacted 51b in the following manner. All objects, classes, fields, methods referenced in a Java class file 24a are identified by using strings in the constant pool 42 of the class file 24a. The card class file converter 26 compacts the constant pool 42 found in the Java class file 24a into an optimized version. This compaction is achieved by mapping all the strings found in the class file constant pool 42 into integers (the size of which is microcontroller architecture dependent). These integers are also referred to as IDs. *Each ID* uniquely identifies a particular object, class, field or method in the application 20.

⁷ Office Action ¶ 2.

Therefore, the card class file converter 26 replaces the strings in the Java class file constant pool 42 with its *corresponding unique ID*.⁸

The clear meaning of the Wilkinson et al. statement “Each ID uniquely identifies a particular object, class, field or method *in the application*” is that each ID (not a combination of IDs) can be used to uniquely identify an object, a class, a field, or a method in the application. See also FIG. 4 of Wilkinson et al., which shows conversion of a class constant pool (reference numeral 42) which “Contains all the strings corresponding to Fields methods and Class names referred to in the Java program” to an optimized card class constant pool (reference numeral 47) where “each string is replaced by *an ID*”.⁹ Since the scope of each ID in Wilkinson et al. is the entire application, the scope is not of limited capacity as recited in amended claim 5. And since there is only one scope (that of the entire application) in Wilkinson et al., there is no teaching of a token type (or optimized numeric values belonging to the same type) as recited in claim 5.

Again, Wilkinson et al. teaches using *a single* token to uniquely identify an item within the entire scope of a software application. In other words, for each item referenced in Wilkinson et al., a single token is used, and each of those tokens has the same scope (that of the entire *application*). Thus in Wilkinson et al. a *single* token may be used to uniquely identify a *class* within the entire scope of the application. Another *single* token may be used to uniquely identify a *field* within the entire scope of the

⁸ Wilkinson et al. col. 9 lines 18-37. (emphasis added)

application. Yet another *single* token may be used to uniquely identify a *method* within the entire scope of the application.

Moreover, the alternative form "OR" as used in the Wilkinson et al. reference cited by the Examiner clearly means that a token can be used for several types of things, but it cannot mean that the same token is used for more than one thing at the same time in the same application because without tokens having scope of limited capacity as disclosed and claimed in the present application, Wilkinson et al. would have different items referenced by the same token. And this would be an unworkable result.

For these reasons, the U.S.C. § 102(e) rejection of claim 5 is unsupported by the art and should be withdrawn.

Independent Claims 1-4, 28-32, and 55-59

Like independent claim 5, independent claims 1-4, 28-32, and 55-59 specify that the corresponding token has a token type and that tokens belonging to the same token type represent the same kind of referenceable item, and where each kind of referenceable item in a package has its own independent scope of limited capacity for tokens of the corresponding token type. Claim 5 being allowable, claims 1-4, 28-32, and 55-59 must be allowable for at least the same reasons.

⁹ Wilkinson et al. FIG. 4. (emphasis added)

Dependent Claims

Claims 7-10 and 16-19 depend from claim 1. Claims 12-15 and 20-21 depend from claim 3. Claims 22-27 depend from claim 5. Claims 34-37 and 43-46 depend from claim 28. Claims 39-42 and 47-48 depend from claim 30. Claims 49-54 depend from claim 32. Claims 61-64 and 70-73 depend from claim 55. Claims 66-69 and 74-75 depend from claim 57. Claims 76-81 depend from claim 59. Claims 1, 3, 5, 28, 30, 32, 55, 57, and 59 being allowable, claims 7-10, 12-27, 34-37, 39-54, 61-64, and 66-81 must also be allowable.

Additionally, claims 7-10, 12-27, 34-37, 39-54, 61-64, and 66-81 recite additional limitations not disclosed by Wilkinson et al.

Claims 7-10, 12-15, 22-23

The Examiner states:

The features of claims 7 10, 12 15, 22 23, see the rejection of claim 5, in view of fig. 4 items 42 and 47. The applicant indicates that Wilkinson does not teach recording in an image of a package a mapping. However, see again col. 5 lines 34 36, lines 45 57 and col. 6 lines 11 16, which indicates that the "the second application has a string of characters and that the string is represented by an identifier (i.e. mapping) in the second application"...¹⁰

Claim 7

Claim 7 recites in part "recording an image of said package a mapping between said token and said referenceable item". The Examiner indicates Wilkinson et al.

¹⁰ Office Action ¶ 2.

indicates "the second application has a string of characters and that the string is represented by an identifier (i.e. mapping) in the second application".¹¹ The Applicants respectfully suggest the Examiner's attempt to equate a mapping with the object of what is being mapped is improper. The claim limitations recite recording a mapping, not recording what is being mapped. Again, contrary to the Examiner's statement, Wilkinson et al. does not disclose recording in an image of a package a mapping between a token and a referenceable item. Rather, Wilkinson et al. discloses a mapping that is separate from the package. More specifically, Wilkinson et al. discloses a card class file converter that receives application class files and a string to ID input map, and outputs a card class file *and a string to ID output map*.¹² For this additional reason, the U.S.C. § 102(e) rejection of claim 7 is unsupported by the art and should be withdrawn.

Claim 8

Claim 8 as amended recites:

The method of claim 1 wherein said referenceable item comprises a class and said reference comprises a package token and a class token.

The Examiner states:

The applicant argues about a single token uniquely identifying an item; while, that is considered the essence of an item having its own independent scope. Types are further inherent in Wilkinson's system to enable different application to access and update the appropriate applications, see figs. 19-20, in a limited resource environment, figs. 21-25.

¹¹ Office Action ¶ 2.

¹² Wilkinson et al. at col. 11 lines 48-58 and FIGS. 3 and 5. (emphasis added)

The Applicants respectfully reiterate that Wilkinson et al. uses a *single* token to uniquely identify an item within the scope of the entire application. Whereas claim 8 specifies a reference comprises a package token *and* a class token (i.e. not a single token). With this Amendment, claim 8 has been modified to make this distinction more clear. Contrary to the Examiner's statement, Wilkinson et al. does not disclose a referenceable item comprises a class and the reference comprises a package token and a class token. Claim 8 specifies the reference comprises a package token and a class token. This is not disclosed by the cited reference. For this additional reason, the 35 U.S.C. § 102(e) rejection of claim 8 is unsupported by the art and should be withdrawn.

Furthermore, the Applicants respectfully submit that such conclusory allegations regarding the alleged inherency of the teachings are improper. According to the MPEP¹³, an Examiner must provide a rationale or evidence tending to show inherency. The fact that a certain result or characteristic may occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic.¹⁴ Furthermore, "In relying upon the theory of inherency, the Examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art."¹⁵

The Applicants respectfully suggest that the Examiner has failed to provide a basis in fact and/or technical reasoning to reasonably support the determination that

¹³ MPEP § 2112.

tokens become class/package tokens necessarily flows from the teachings of Wilkinson et al.

Nowhere in Wilkinson et al. does it disclose that a referenceable item comprises a class and a reference comprises a package *and* a class token. For this additional reason, the U.S.C. § 102(e) rejection of claim 8 is unsupported by the art and should be withdrawn.

Claims 9-10

Claim 9 as amended recites:

The method of claim 1 wherein said referenceable item comprises a field and said reference comprises a package token, a class token, and a field token.

Claim 10 as amended recites:

The method of claim 1 wherein said referenceable item comprises a method and said reference comprises a package token, a class token, and a method token.

Wilkinson et al. does not disclose a referenceable item comprising a field and a reference comprising a package token, a class token, and a field token. Nor does Wilkinson et al. disclose a referenceable item comprising a method and a reference comprising a package token, a class token and a method token. Again, as mentioned above, Wilkinson et al. teaches using a *single* token to uniquely identify an item within the entire scope of a

¹⁴ *In re Rijckaert*, 9 F.3d 1531, 1534, 28 USPQ2d 1955, 1957 (Fed. Cir. 1993).

¹⁵ *Ex parte Levy*, 17 USPQ2d 1461 (Bd. Pat. App. & Inter. 1990) (emphasis in original).

software application. For this additional reason, the U.S.C. § 102(e) rejection of claims 9 and 10 is unsupported by the art and should be withdrawn.

Claims 12-15

Claims 12-15 recite limitations similar to claims 7-10. Claims 7-10 being allowable, claims 12-15 must also be allowable.

Claim 22

Claim 22 recites:

The method of claim 5 wherein said package further comprises at least one reference to an internal item.

Wilkinson et al. does not distinguish between internal items and those exported or otherwise made available to other applications with respect to token assignment. Rather, Wilkinson et al. teaches merely assigning tokens to constant pool entry. A reference consisting at least in part of an offset to an actual item is not mentioned by Wilkinson et al.

The Applicants respectfully suggest that the Examiner has failed to provide a basis in fact and/or technical reasoning to reasonably support the determination that the package comprises at least one reference to an internal item as claimed in claim 22 necessarily flows from the teachings of Wilkinson et al. For this additional reason, the U.S.C. § 102(e) rejection of claim 22 is unsupported by the art and should be withdrawn.

Claims 23-27 depend from claim 22 and thus include the limitations of claim 22. The base claim being allowable, the dependent claims must also be allowable for at least the same reasons.

Claim 23

Claim 23 recites:

The method of claim 22 wherein said internal item comprises a class and said reference comprises an offset within said package to a class record associated with said class.

The Applicants respectfully suggest that the Examiner has failed to provide a basis in fact and/or technical reasoning to reasonably support the determination that the internal item comprises a class and that the reference comprises an offset within the package to a class record associated with the class as claimed in claim 23 *necessarily* flows from the teachings of Wilkinson et al. For this additional reason, the U.S.C. § 102(e) rejection of claim 23 is unsupported by the art and should be withdrawn.

Claims 19 and 21

Claim 19 recites:

The method of claim 18 wherein said indication of said implementation of said interface method comprises an index into a virtual method table.

Claim 21 recites:

The method of claim 20 wherein an indication of said implementation of said interface method comprises an index into a virtual method table.

The Examiner states:

In reference to claims 19 and 21, see again the string (virtual) to ID (real) mapping above. Also, see col. 14 lines 41-66.¹⁶

The portion of Wilkinson et al. referenced by the Examiner recites:

To enforce the security model of the programming language, a 256-byte table is created as shown in Appendix G which is hereby incorporated by reference. This table is indexed by the byte code number. This table contains the type and length information associated with the indexing byte code. It is encoded with the first 5 bits representing type, and the last 3 bits representing length. The type and length of the byte code is indexed directly from the table by the byte code number. This type and length is then used for checking as shown in Appendix H which is hereby incorporated by reference. In Appendix H, the checking process begins by decoding the length and type from the table in Appendix G which is hereby incorporated by reference. The length is used to increment the program counter. The type is used first for pre-execution checking, to insure that the data types on the VM stack 144a are correct for the byte code that is about to be executed. The 256 bytes of ROM for table storage allows the original Java byte codes to be run in the Card JVM 16 and minimizes the changes required to the Java class file to be loaded in the card. Additional Java byte codes can be easily supported since it is relatively easy to update the appropriate table entries.

In other embodiments, as shown in FIG. 10, the Java byte codes in the method are renumbered in such a manner that the byte code type and length information stored in the table in Appendix H is implicit in the reordering.¹⁷

Thus, the reference provided by the Examiner relates to a 256-byte table used to enforce the security model of a programming language. The table is indexed by a byte code number and contains type and length information associated with the indexing byte

¹⁶ Office Action ¶ 2.

¹⁷ Wilkinson et al. col. 14 lines 41-66.

code. The table in Appendix G of Wilkinson et al. has nothing to do with determining which set of bytecodes (method) to execute. More specifically, the passage pointed to by the Examiner makes no reference to an index into a virtual method table as claimed in claims 19 and 21. Whereas the interface method table and virtual method table designed by the applicants contain information for easily determining which interface method to execute. The Examiner is reminded that in a rejection under 35 U.S.C. § 102, the identical invention must be shown in as complete detail as contained in the claim.¹⁸ For this additional reason, the U.S.C. § 102(e) rejection of claim 19 and 21 is unsupported by the art and should be withdrawn.

Dependent Claims 34-37, 39-54, 61-64, and 66-81

Dependent claims 34-37 and 61-64 include limitations similar to claims 7-10. Claims 7-10 being allowable, claims 34-37 and 61-64 must also be allowable. Additionally, dependent claims 39-54 and 66-81 include limitations similar to claims 12-27. Claims 12-27 being allowable, claims 39-54 and 66-81 must also be allowable. Accordingly, the Applicants respectfully request the U.S.C. § 102(e) rejection of claims 34-37, 39-54, 61-64, and 66-81 be withdrawn.

Accordingly, it is respectfully requested that the 35 U.S.C. § 102(e) rejection of claims based on Wilkinson et al. be withdrawn. In view of the foregoing, it is respectfully asserted that the claims are now in condition for allowance.

¹⁸ *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913,1920 (Fed. Cir. 1989).

Request for Allowance

It is believed that this Amendment places the above-identified patent application into condition for allowance. Early favorable consideration of this Amendment is earnestly solicited.

Request for Entry of Amendment

Entry of this Amendment will place the Application either in condition for allowance, or at least, in better form for appeal by narrowing any issues. Accordingly, entry of this Amendment is appropriate and is respectfully requested.

Appl. No. 09/243,108
Amdt. dated December 2, 2003
Reply to Final Office Action of July 2, 2003

Docket No. SUN-P3730
(811173-000186)

If, in the opinion of the Examiner, an interview would expedite the prosecution of this application, the Examiner is invited to call the undersigned attorney at the number indicated below.

Respectfully submitted,
THELEN REID & PRIEST, LLP

Dated: December 2, 2003



John P. Schaub
Reg. No. 42,125

Thelen Reid & Priest LLP
P.O. Box 640640
San Jose, CA 95164-0640
Tel. (408) 292-5800
Fax. (408) 287-8040-